Amazon KDD Cup 2024: Multi-Task Online Shopping Challenge for LLMs report

하지원
서강대학교
컴퓨터공학과

Matti Zinke
서강대학교
컴퓨터공학과

심규재
서강대학교
아트엔테크놀로지학과

Abstract

우리 Ensemble팀은 KDD CUP 2024에 참여하여 Round 1에 참가한 130 팀 중에서 Round 2에서 23위, Score는 0.689를 기록하였다. 이는 프롬프트 엔지니어링과 20 종의 pre-trained LLM에 대한 사용 시도를 통해 얻어진 결과이다. 그중에서 가장 성능이 높았던 LLM은 DPO 또는 DARE-TIES 기술을 기반으로 하며, GSM8K 성능지표와 비례하는 상관성을 보인다.

1. Introduction

Competition Overview

세계적인 데이터마이닝 학회 SIGKDD에서 개최하는 데이터마이닝 경연대회인 KDD CUP은 올해 아마존 사의 주관으로 진행 중이다. 아마존 사에서 제공하는 데이터셋을 기반하여 LLM을 학습시키고, (1)Shopping Concept Understanding, (2)Shopping Knowledge Reasoning, (3)User Behavior Alignment, (4)Multilingual Abilities, (5)All-Around 이 다섯 가지 Task를 해결하는 것이 이 대회의 목표이다. 본 팀이 참여한 Track 1은 Shopping Concept Understanding task를 다루며, 상품명, 상품 카테고리, 속성, 상품 설명, 구매 후기 등 텍스트 형태로 주어진 데이터를 LLM이 얼마나 이해하는 가를 평가한다.[1]

[1] Amazon Search, “Amazon KDD Cup 2024: Multi-Task Online Shopping Challenge for LLMs-Track 1: Shopping Concept Understanding”, Aicrowd.com, April 2024. <https://www.aicrowd.com/challenges/amazon-kdd-cup-2024-multi-task-online-shopping-challenge-for-llms/problems/amazon-kdd-cup-24-understanding-shopping-concepts> (2024.06)

Dataset

실제 아마존의 쇼핑데이터에서 추출된 57개의 task와 20,598개의 질문을 가진 ShopBench 데이터셋으로 부터 다시 19개의 task와 96개의 질문을 추출한 데이터셋인 ‘development.json’이 참가자들에게 제공된다. 아래는 ‘development.json’의 질문 데이터 중 하나다.

{
"input_field": "Instructions: Tell me what this product category is about\nInput: Toggle Switch\nOutput:",
"output_field": "A toggle switch is an electric switch operated by means of a projecting lever that is moved up and down.",
"task_name": "task1",
"task_type": "generation",
"metric": "sent-transformer",
"is_multiple_choice": false,
"track": "amazon-kdd-cup-24-understanding-shopping-concepts"
}

‘input_field’는 사용자가 입력하는 text이며, 지시사항 및 문제 내용을 담고있다. ‘output_field’는 시스템이 출력할 목표 text의 쌍인 정답 내용을 담고 있다. ‘task_name’은 KDD CUP이 제시한 5가지 task 중 어느 유형인지를 나타낸다. ‘metric’은 해당 문제의 성능 지표를 담고있다. ‘is_multiple_choice’는 여러 선지를 제공하는 task에서 적용되는 조건이고, 해당 문제가 단답형인지 확인하는 목적이다. ‘track’은 어느 트랙의 문제인지 명시한다.

Model Overview

이 프로젝트에서 쓰일 모델의 입출력 조건은 다음과 같다. 주요 변수를 먼저 살펴보자면, LLM에 입력할 프롬프트를 입력받을 텍스트 타입의 'prompt' 변수, 그리고 해당 문제가 선다형(multiple choice) 문제인지 아닌지를(True or False) 구분하는 bool 타입의 'is_multiple_choice' 변수가 있다. 정확히는 모델 class의 내부 메소드인 'predict'의 매개변수들이며, 이 함수의 반환값은 문자열이다. 선다형 문제에 대해서는 선택사항이 4가지인 관계로 0에서 3사이의 정수 값 중 하나가 무작위로 출력된다. 혹은 Ranking과 Retrieval task를 위한 정수, Named Entity Recognition tasks를 위한 명칭(named entities), generation task를 위한 생성된 답변(generated response) 중 하나를 대변하는 문자열이 답변으로 출력한다. 모델을 구성하기 위해서는 'base_model.py'의 'ShopBenchBaseModel' class를 디자인해야 한다. 'dummy_model.py'의 예시를 보면 선다형 문제인지 아닌지에 대해서만 출력하도록 되어있는데, 문제가 선다형일 경우, 'possible_responses' 리스트의 1부터 4까지의 정수 중 하나가 무작위로 문자열로 변환되어 출력되며, 선다형이 아니라면 'possible_responses' 리스트를 무작위로 셔플링한 결과를 출력하도록 되어있다. 이제 'prompt' 변수를 어떻게 받아들이고 처리할 지를 구현해야한다.

2. Methodology

주최 측에서 기본 모델로서 'Dummy model'을 제공한다. 학습 과정 없이 랜덤하게 결과를 출력하는 'DummyModel'의 경우, 'local_evaluation.py'를 통해 성능을 검증해본 결과, 0.15199378603996178가 나왔다. 이보다 더 높은 정확도를 얻으려면 'development.json'을 few-shot 학습할 수 있는 적절한 LLM 선택과 프롬프트 엔지니어링이 필요하다. 처음 시도된 LLM은 주최 측의 baseline notice에 명시된 Vicuna-7b이다. 이 모델은 LLaMA에 기반하여 파인튜닝된 챗봇이다. 프롬프트 입력 방식은 'task_name'에 따른 구분 없이 동일한 내용의 general system prompt가 'input_field'와 합쳐져 입력되는 방식으로 세팅되어 있다. 아래는 프롬프트의 내용이다.

general system prompt=

> "You are a helpful online shopping assistant. Please answer the following question about online shopping and follow the given instructions.\n\n"

위 프롬프트로 작동한 Vicuna-7b 로부터는 만족스러운 결과를 얻지 못하였다. 성능 향상을 위해 아래와 같이 시스템 프롬프트를 수정하여 진행했다.

general system prompt (modified) =

> "Please listen to the given task carefully and pay special attention on how the answers should be give.\n"

위 프롬프트 뿐 아니라 여러 프롬프트에 대해서도 유의미한 성능 향상이 발생하지 않았기에, 모든 task에 적용되는 general system prompt 방식 자체의 문제라 판단했다. 따라서 조건문으로 특정 task를 식별할 수 있으면 specific system prompt를 사용하되, 그렇지 않을 경우는 default로 general system prompt를 사용하도록 구현했다. 아래는 specific system prompt 중 일부의 예시이다.

Task #1 specific system prompt =

> "You have to explain the given product category or type. Your answer should be in one sentence of a length of about 20 words."

Task #7 specific system prompt=

> "You will be given a list and a specific review. You should EXACTLY choose 3 aspects as numbers from the given list that are covered in the review and nothing else."

Task #17 specific system prompt =

> "You will be given a product title in inverted commas. You have to translate it based on the language given in the description. You should not explain the answer."

이러한 task별 프롬프트 방식으로 기존 방식 대비 성능향상을 확인 할 수 있었다.

3. Results

여러 LLM중 어느 모델이 이러한 task별 프롬프트 방식으로 성능이 높을지는 'local evaluation.py' 실행 전에는 알 수 없으므로, Hugging Face에서 다양한 오픈소스 LLM을 불러와서 성능을 확인하고, 파인튜닝하는 것이 바람직하다고 판단했다. 총 20개의 LLM에 대해 정확도 산출을 진행했으며 그 결과는 아래의 표와 같다.

LLM 모델명	local eval score
DARE_TIES_13B	0.5956676126566732 → 0.6213757661369984
13B_MATH_DPO	0.6148088357017373
Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B	0.6140449173494128
MixTAO-7Bx2-MoE-v8.1	0.5896902518997831
MoE_13B_DPO	0.5788059831556638
cr-model-v1	0.577217490607002
Mistroll-7B-v2.2	0.5682951110016166
Qwen1.5-7B-sft-0502	0.5610776246185725
CarbonBeagle-11B-truthy	0.5593031144819792
Experiment26-7B	0.5280649422643182
T3Q-EN-DPO-Mistral-7B	0.5261771492540045
Meta-Llama-3-8B-Instruct	0.5223284217275423
Calme-7B-Instruct-v0.2	0.5153258876929895
Hermes-2-Theta-Llama-3-8B	0.5125085712022551
Qwen1.5-14B	0.4997765156772354
stablelm-2-12b-chat	0.46852982312194846
Qwen1.5-MoE-A2.7B-Chat	0.45975688578281115
Yi-9B	0.4544301238436799
Qwen-14B-Llamafied	0.41060220095147437
Yi-6B	0.30425252628858446

이 20개 모델 중 정확도가 가장 높았던 3가지를 선정하면 DARE_TIES_13B, 13B_MATH_DPO, Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B이다. 이 중에서 정확도가 약 60%가 나왔던 DARE_TIES_13B에 대해 Task 2 내용을 기반으로 일부 프롬프트를 수정하였더니 약 62%로 개선되었다. 최종적으로는 해당 모델이 가장 높은 성능을 보였으며, 이 모델을 기반으로 한 코드를 제출하여 Round 2에서 Leaderboard 23위, Score는 0.689를 기록하였을 뿐 아니라 상대적으로 Runtime도 적게 소요되었다.

Analysis between local eval score - LLM Performance Indicator

우리 팀에서 시도한 20가지 오픈소스 LLM은 모두 HuggingFace에서 자체적으로 평가한 0~100 사이의 성능지표를 가지고 있다.[2] 성능지표에는 ARC, HellaSwag, GSM8K 등이 있으며 이를 종합한 average 값이 제공된다. 이에 팀에서 얻어낸 모델별 'local eval score' 값과 성능지표들의 수치를 비교분석 하는 것으로 특정 성능지표가 정확도에 기여하는 비중을 추론할 수 있다고 가정했다. 이에 7가지의 LLM 성능지표와 'local eval score'의 값에 대해 상관분석과 회귀분석을 진행하였으며 그 결과 값은 아래와 같다.

[2] Edward Beeching et al. ,"Open LLM Leaderboard", Hugging face, 2023 <https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard> (2024.06)

Model name- Performance Indicator Table

LLM name\ Performance Indicator	average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	localscore
DARE_TIES_13B	77.1	74.32	89.5	64.47	78.66	88.08	67.55	0.621375766
yunconglong/13B_MATH_DPO	77.08	74.66	89.51	64.53	78.63	88.08	67.1	0.614808836
yunconglong/Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B	77.44	74.91	89.3	64.67	78.02	88.24	69.52	0.614044917
zhengr/MixTAO-7Bx2-MoE-v8.1	77.5	73.81	89.22	64.92	78.57	87.37	71.11	0.589690252
yunconglong/MoE_13B_DPO	77.05	74.32	89.39	64.48	78.47	88	67.63	0.578805983
TwT-6/cr-model-v1	77.32	70.65	87.85	74.73	80.47	83.66	66.57	0.577217491
BarraHome/Mistroll-7B-v2.2	76.76	72.78	89.16	64.35	78.1	85	71.19	0.568295111
Qwen1.5-7B-sft-0502	61.99	55.12	77.18	61.68	50.72	71.67	55.57	0.561077625
vicgalle/CarbonBeagle-11B-truthy	76.1	82.27	89.31	66.55	78.55	83.82	66.11	0.559303114
yam-peleg/Experiment26-7B	76.64	73.88	89.15	64.32	78.24	84.93	70.43	0.528064942
chihoonlee10/T3Q-EN-DPO-Mistral-7B	76.43	73.04	89.3	64.13	78.71	85.32	68.08	0.526177149
Meta-Llama-3-8B-Instruct	66.87	60.75	78.55	67.07	51.65	74.51	68.89	0.522328422
MaziyarPanahi/Calme-7B-Instruct-v0.2	76.61	73.12	89.19	64.36	78	84.93	70.05	0.515325888
Hermes-2-Theta-Llama-3-8B	68.1	66.04	84.95	63.36	55.75	78.06	60.42	0.512508571
Qwen/Qwen1.5-14B	66.7	56.57	81.08	69.36	52.06	73.48	67.63	0.499776516
stablelm-2-12b-chat	68.38	64.85	85.96	61.06	62.01	78.53	57.85	0.468529823
Qwen1.5-MoE-A2.7B-Chat	57.22	53.67	80.54	60.97	50.56	69.38	28.2	0.459756886
01-ai/Yi-9B	63.17	61.18	78.82	70.06	42.45	77.51	48.98	0.454430124
Qwen-14B-Llamafied	63.09	55.2	82.31	66.11	45.6	76.56	52.77	0.410602201
01-ai/Yi-6B	54.08	55.55	76.57	64.11	41.96	74.19	12.13	0.304252526

Correlation Analysis Table

	average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	localscore
average	1							
ARC	0.922854	1						
HellaSwag	0.930937	0.914986	1					
MMLU	0.177081	0.071087	-0.01644	1				
TruthfulQA	0.950588	0.924376	0.947515	0.039973	1			
Winogrande	0.920184	0.923232	0.91484	0.090632	0.901755	1		
GSM8K	0.862531	0.67405	0.687896	0.223539	0.701635	0.652652	1	
localscore	0.816958	0.699199	0.680244	0.077637	0.769455	0.675213	0.807009	1

Regression Analysis Table

조정된 결정계수	0.653715
표준 오차	0.045697
관측수	20

	자유도	**제곱합**	**제곱 평균**	**F 비**	**유의한 F**
회귀	7	0.08952	0.012789	6.124007	0.003254
잔차	12	0.025059	0.002088		
계	19	0.114579			

	계수	**표준 오차**	**t 통계량**	**P-값**	**하위 95%**	**상위 95%**	**하위 95.0%**	**상위 95.0%**
Y 절편	1.094865	0.551996	1.983467	0.070673	-0.10783	2.297561	-0.10783	2.297561
average	0.001916	0.056407	0.033963	0.973465	-0.12098	0.124816	-0.12098	0.124816
ARC	-0.00049	0.005062	-0.09649	0.924722	-0.01152	0.010542	-0.01152	0.010542
HellaSwag	-0.01206	0.011697	-1.03116	0.322802	-0.03755	0.013424	-0.03755	0.013424
MMLU	-0.00311	0.009207	-0.33756	0.741527	-0.02317	0.016952	-0.02317	0.016952
TruthfulQA	0.004677	0.010528	0.444257	0.664763	-0.01826	0.027615	-0.01826	0.027615
Winogrande	0.001078	0.015964	0.067554	0.947253	-0.0337	0.035861	-0.0337	0.035861
GSM8K	0.002707	0.009745	0.277812	0.785885	-0.01852	0.023939	-0.01852	0.023939

상관 분석 결과 'local eval score'과 GSM8K간의 정비례 상관성을 확인할 수 있었다. GSM8K가 중학교 수준의 수학문제들에 대한 데이터셋임을 고려했을때[3] 단계별 추론능력과 연산능력 등을 갖춘 LLM이 성능이 높을 것이라 추론할 수 있다. 회귀 분석 결과 조정된 결정계수와 F값은 유효하나 p-값이 0.005이하인 성능지표가 없다는 점에서 유의미한 결과를 찾지 못했다.

[3]Cobbe et al. ,"OpenAI/grade-school-math", GitHub, 2021 <https://github.com/openai/grade-school-math > (2024.06)

5. Conclusion

'local eval score'값이 가장 높았던 모델 3가지는 DPO 및 DARE-TIES 기술 기반의 모델이었다. DARE_TIES_13B 모델에 대한 프롬프트 엔지니어링 방식을 일부 수정해봤더니 최종 성능이 약 0.65로 개선됨을 확인할 수 있었다. 하지만 본 팀은 단순히 공개된 LLM을 불러와 주어진 데이터셋만으로 추론을 한 것에 불과하다. 따라서 기존 방식으로 얻은 성능 결과를 어떻게 더 높일 수 있을지 향후 계획으로서 다음과 같이 5가지 가설을 세워봤다.

Future Goal

첫 번째는 development.json 내용을 생성형 LLM에 학습시켜 추가적인 데이터셋을 생성한후, Transfer Learning시키는 것이다.
두 번째는 API를 통해 아마존 쇼핑 웹사이트에 접근하여 실제 존재하는 정보를 들여와 결과값과 비교하는 것이다.
세 번째는 지금보다도 더 정확한 답을 유도할 수 있는 시스템 프롬프트를 찾는 것이다. 이를 위해 선다형 질문을 필터링하면서 주어진 프롬프트의 목표를 더 자세히 분석해볼 수 있다.
네 번째는 ECInstruct, Amazon-M2등의 데이터셋을 이용해 fine-tuning 하는 것이다.
다섯 번째는 DPO, DARE-TIES기술을 기반으로 하거나 GSM8K 성능지표가 뛰어난 오픈소스 LLM을 선택 또는 직접 이러한 LLM을 구현해보는 것이다.
이 다섯 가지 가설을 시도해봄으로써 모델 성능을 개선해볼 것이다.

Reference

[1] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, Yongbin Li et al. Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. *arXiv preprint arXiv:2311.03099*, 2023

[2] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, Mohit Bansal et al. TIES-Merging: Resolving Interference When Merging Models. *arXiv preprint arXiv:2306.01708*, 2023

[3] Rafael Rafailov, Archit Sharma, Eric Mitchell, et al. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint arXiv:2305.18290*, 2023

[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*, 2022

[5] Edward J. Hu, Yelong Shen, Phillip Wallis et al. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2105.09685*, 2021